# Markup Systems in the Present

Steven J. DeRose
Electronic Book Technologies

## Introduction

In our 1987 article "Markup Systems and the Future of Scholarly Text Processing," reprinted elsewhere in this volume, James H. Coombs, Allen H. Renear, and I presented a theoretical framework for understanding systems for marking up documents. We defined *presentational, procedural,* and *descriptive* markup methodologies and argued for the superiority of descriptive markup. Our arguments were based on perspicuity and consequent portability; minimization of cognitive demands for the user; flexibility of interpretation, application, and processing; and a variety of benefits of standardization per se.

Now, five years later, a number of historical developments have affected the movement toward the future we described. In this article I shall survey progress and major trends in markup systems and discuss likely future directions and issues.

Descriptive markup is the practice of categorizing and labeling portions of documents in terms of their functional roles, rather than in terms of their intended processing. For example, a section title would be marked up as such, rather than as "Helvetica 18-point bold." Obviously the first characterization is useful for a wider range of purposes than the second; it captures an essential quality that the second neglects, and (perhaps most importantly) it is directly related to the author's model of a document, rather than a typesetter's.

In 1987 we described three major problems arising from the historical shift of emphasis in text-system design from modeling the user as an author to modeling the user as typist, typesetter, or programmer-typist. Those problems were

1. Decreased awareness of the significance of the document's electronic form in its own right

2. Decreased incentive for research in advanced document-creation systems for authors

3. Distraction of authors from their primary tasks of research and composition toward managing typesetting and page-layout details.

In general, the first problem has been largely overcome, and this success is gradually leading to resolution of the second. The third, however, remains.

### State of the Art

It has now become possible to handle structured electronic documents with much more facility than in the past. Portability of raw data has been nearly, though not fully, achieved. Tools for composition are more available, if still not advanced. However, software only incompletely supports the life cycle of electronic documents because electronic submission to publishers and electronic document delivery are not yet simple or widespread.

#### *Portability*

The need for portability of documents has been continually increasing. Paper documents' strongest advantages over electronic ones center on portability and compatibility. Many causes have contributed to an increased consciousness of the need for comparable portability of electronic documents and manuscripts and thence to consciousness of the advantages of descriptive, structure-centered markup. These causes include

1. Drastic increases in the cost of paper

2. Increased access to wide-area networks such as the Internet

3. Increasing internationalization of the network community

4. Increasing availability of major on-line text collections of interest to researchers, particularly in the humanities.

Such factors have led to growing awareness that having a different proprietary format for each person's favorite word processor—or each person's favorite printer, fonts, and paper size—is maladaptive. Users have begun to value the ability to pass electronic documents around transparently, as they have been doing for centuries with paper documents.

Strategies for achieving portability have been myriad. At the lowest level, networked users must still employ special strategies just to move bytes without loss, since nodes on the Internet currently disagree about how and whether to forward both eight-bit characters and some seven-bit characters such as braces and brackets, backslashes, and exclamation points.[1]

At the next level, software vendors still build multiple Save As Format X options for their most popular competitors' formats. These features, now as then, break with each new software version. A subtler problem is that even when the input document encodes some structural information (say, by style sheets), such information seldom survives the transfer: conversion is usually limited to producing a similar page layout. Many users give up and mail page images instead of documents, either via PostScript or bitmaps, or discard all but the most rudimentary presentational markup, calling the result "plain ASCII." Such representations neither represent nor even encourage the recovery of structural information. Similarly, no consensus on graphics formats seems to be present, and there is not even an analog to "plain ASCII" as a fallback.

For network communications, a variety of presentational markup methods are typically used, such as blank lines, indentation, and conventional meanings for strings such as `> (quotation)`, `* (emphatic)`, and even `;) (joke)`. Unfortunately, these methods are neither rigorous nor unambiguous and have limited capacity for expressing document structure.

In communities with large document communication needs, or with many levels of co-authors working together, the problems of interchange are felt especially acutely. Particularly in such environments, Standard Generalized Markup Language (SGML) has become the medium of choice for document interchange, and therefore, tools to convert into and out of it as an "interlingua" have proliferated. Such tools are only beginning to be accessible to most users.

*Writing Tools*

Tools that cater to writers' needs as opposed to typesetters' and graphic designers' needs are still rare. Page-layout tools are much more sophisticated, but the lamented regression from cognitively appropriate markup has not been entirely reversed. Useful structural tools such as outline processors also continue to improve but remain in isolation from other tools. It remains difficult to treat the same document as a source for publication and for outline processing, presentation management, information retrieval, on-line delivery, and other uses.

Three trends, however, give hope that writer-oriented tools employing document models analogous to document structures will grow in availability and use. First, style sheets are now an expected feature of serious word processors of all kinds. Style sheets encourage at least a rudimentary level of descriptive markup, although such markup is not rigorous (for example, software that makes it easy to create new styles when needed tends also to make it easy to create redundant or ill-conceived styles when such are not needed).

Second, recent emphasis on object-oriented design has encouraged software designers to think of data structures in terms of classes, subclasses, and instances. This approach contributes to acceptance of analogous structured document models such as SGML, and research in object-oriented design provides enabling methodologies for building effective structured document systems. Briefly put, a kind of document element such as a chapter, bibliography entry, or emphatic phrase is comparable to a class. An instance of such an element must have internal structure conforming to the class definition, and SGML-element instances may have attribute lists, comparable to instance variables. The analogy, though useful, is not perfect: although class hierarchies typically represent relations of type and subtype, for documents the more salient hierarchy is one of containment. Tools that directly manipulate these hierarchical structures have now become available to authors.

Third, a number of products have begun to add processing capabilities to documents via "scripting languages" that provide for interaction between the user and document portions. Although such systems as yet lack an advanced model of document structure, they open the door to processing a single document in various ways for various purposes.

Much as with the advent of structured programming, those managing very large projects feel most acutely the need for a rigorous and abstract approach. Hence, two groups have been particularly quick to adopt descriptive markup and create major documents and collections using it: high-tech industries, which have enormous manuals, and humanities scholars, who work closely with large collections of literary texts.

*Electronic Submission*

Publishers were early supporters of descriptive markup, presumably because of the potential economic advantage gained if authors submit usable manuscripts in electronic form. The Association of American Publishers (AAP) developed one of the first industry-standard applications of SGML, and it has now become an

American National Standards Institute (ANSI) standard. However, few publishers yet permit electronic submissions, and rekeying of manuscripts is still the norm except for authors willing to assume the entire responsibility for producing camera-ready final copy.

The ubiquity of Internet access makes authors increasingly reluctant to accept delivery delays and postal costs for manuscripts. An interim solution adopted by many publishers has been to accept manuscripts in one or two word-processor formats, or "plain ASCII," the definition of which varies. The latter option saves rekeying text but foregoes representation not only of document structure but even of formatting. It also fails to represent basic nonlinearities such as footnotes and cross-references, as well as document portions with complex structures such as tables, graphics, and equations.

*Electronic Delivery*

Electronic document delivery has not yet arrived, although a few systems do exist. Sony has produced the Data DiscMan, a hand-held device that can display documents from an optical disk several lines at a time. Microsoft has published many documents on line in its Bookshelf series. Owners of large data and statistics collections frequently sell them in electronic form.

The present situation, however, is that these products provide only minimal functionality. In almost all cases the data can be read, searched, and printed—but only by proprietary software sold with it, and only on specific hardware and operating system versions. Such specific software cannot anticipate all legitimate uses for the data, and so users with novel applications are left out. In addition, the formatting capabilities of such systems seem primitive even when compared to low-end word processors.

A few large projects in academia and industry are moving toward delivering extremely large documents on line, again because such documents present the need most acutely. The US Department of Defense has continued with the CALS initiative, and is working toward an Interactive Electronic Technical Manual system much like those prototyped in the seventies (Feiner). Scholars have also moved in this direction, as the primary literatures of many fields are being encoded and distributed, and text archives around the world gradually convert to common formats.

Theodor H. Nelson's vision has yet to be achieved: "We must leave paper behind at last if we are to move to the new worlds of tomorrow. And that includes leaving

behind such last-ditch anachronisms as 'Wizzywig' and 'modelessness.' These all have their eye on the past.... Today's 'computer literacy' and styles of operation will seem to us then like poking the cat's-whisker on the crystal in the radio set of 1920. 'It's what we had to do then,' we'll say" (24).

*Factors Slowing Adoption of Descriptive Markup*

Working against these encouraging developments there remain two major technical issues and one major economic one, all of which have yet to be resolved before all the benefits of descriptive markup can be realized. First, there remains a concern (sometimes legitimate and sometimes fatuous) for "page fidelity": keeping line breaks, page breaks, and text geometry constant across transmission. It is still widely thought that a page-description method such as PostScript will address this issue, but such solutions are limited because (1) the text content and structure are not well represented, making any processing other than printing difficult, and (2) even printing another's PostScript or similar document is incompletely reliable because of incompatibilities of fonts, printer resolution, and so on.

In the realm of legal publishing, and for certain other very particular applications, exact page fidelity may be needed. However, the usefulness even of current legal databases that do essentially no formatting is great, so this concern is clearly limited. Also, even highly format-oriented software often fails to produce *exactly* the same layout and appearance, owing to factors such as round-off error in calculating page placements and character widths and a proliferation of program options that cannot be automatically synchronized when a file is transferred.

The second technological factor slowing adoption of structured and portable documents is the ubiquity of the facsimile machine. Despite poor image quality and the fact that not even the individual characters of a document can be easily recovered from a facsimile's bitmap result, fax machines are so convenient and so standardized that they decrease the perceived need for processable representations.

Economically, issues of copyright plague electronic texts as they do many other areas of computer science.

## New Capabilities

As the number of electronic texts grows, so does the need for tools with which to process and present them. The areas most under development now include structured search, custom-tailored display and rendering, and interactive documents, commonly treated under the heading "hypermedia."

As we described the situation in 1987, structured representation encourages processing document portions by class. For example, a logician may require special handling of axioms, expressions, and proofs, whether for formatting, indexing, or even validation by expert systems. Research in the particular problems of search in structured documents is only beginning.

The central theoretical problem is the notion of containment. When it was sufficient to model documents as unitary objects with minimal internal structure, retrieval methods employed criteria such as how many times a document contained a word, contained two words near each other, and so on. Most systems provided a single level of granularity, typically either "document" or "paragraph," but not both, and certainly not an unlimited number of named levels with user-defined significance.

As large documents with complex structure have come into use, programmers often use two strategies to recycle available retrieval methods: first, documents are restructured to fit the available methods, fragmented into topical portions that become "documents" in their own right. This strategy is problematic for two reasons: pre-existing documents must be changed, and new documents must be written in a new staccato style so that information prerequisite to a fragment is duplicated in the fragment or referred to. Such style has some advantages and imposes a perhaps desirable discipline on authors; however, it is prescriptive and precludes presentation in a flow or sequence of greater size when the topic or the author's rhetorical style justifies it.

The second strategy is to choose a small number of the more tractable document elements and index them specially. This approach almost always builds upon a relational database model and may even refer to document elements as fields. As a consequence of the relational model, the indexers choose elements that tend not to repeat within a document: author, publication date, and similar front matter. Such models fail to describe the structure of real documents, in which very few fields occur only once or have no significant internal structure (of the 210-plus element types defined in the AAP tag set, only 15 can have only a limited number of subelements).

Relational implementations for searching hierarchical documents are possible; indeed many exist. However, by definition a relational element ("record") cannot provide for a variable number of subelements ("fields"); thus, such an implementation must force-fit document structure into an inappropriate model, which is

ineffective for large documents or document collections. An adequate system must associate text content with multiple containers, not just one: a word occurs not only in its paragraph but in every higher containing element, up to the book itself. Only a handful of systems treats this problem seriously.

Recent efforts such as those of Forbes J. Burkowski at the University of Waterloo in developing text algebras for predication of containment and precedence relations, and treatments of similar structures developed by theoretical linguists, promise to lead to more sophisticated tools in this area.

*Dynamic Display*

If a markup system expresses in documents information only about document layout and formatting, then it naturally will express only one such layout. However, descriptive markup's separation of formatting from structural information opens the possibility of associating any number of layouts with a single document. Similarly, if the paper page is definitive, only one form makes sense; but if a changeable medium such as a screen is the document's destination, an unlimited number of specialized layouts should be possible. Tools that format documents on the fly have occasionally been developed in the past, and most current word processors can do this vertically for small documents (although few permit even so simple a horizontal operation as rebreaking text to fit the window).

The capabilities of early systems such as FRESS (Carmody, Catano, Prusky) to hide structural document elements selectively, and to change display styles at the user's whim, have only now begun to reappear. Excessive focus on page layout has led virtually all presentation systems to implement serial formatting algorithms, which are in principle not capable of adjusting rapidly to user actions that globally affect large documents.

Structure-aware tools permit readers to customize documents to their needs, such as by hiding information required only for experts or novices, or showing only the chapters of a document that contain certain content or have certain attributes. Database viewers support all these operations but still have display quality too impoverished for dealing with documents.

*Hypertext/Hypermedia*

Many of the topics already discussed impinge on hypertext and hypermedia, topics that have grown since the first major conference (Hypertext '87) to become required

check-box items for new software and which have in the process lost some of their meaning.[2] Hypertext-related software provides a means to journey from here to there at the click of a button. However, the natures of "here" and "there" themselves are commonly neglected, and documents are often constrained by a model even less revealing than those of format-oriented word processors: the card.

The earliest hypertext systems were closely related to the earliest descriptive markup systems and consequently treated documents under much more sophisticated models. Augment (Engelbart) provided a work environment where substantial documents as well as day-to-day electronic mail were created, edited, and linked. Brown University's FRESS provided for intermixing hierarchical and nonhierarchical structures, associating keywords with document elements and preserving links to all these across editing while supporting documents as diverse as poetry, textbooks, and faculty-salary databases. Nelson, who contributed to the development of FRESS, calls it "the first visual word processor" (15), and it was also the first hypertext system used in teaching. Scribe, perhaps the earliest pure descriptive markup system, grew to become HyperScribe (see Reid).

However, at the same time that the popularization of WYSIWYG word processing was accompanied by proliferation of inadequate document models based upon the typewriter, the popularization of linking was accompanied by proliferation of inadequate hyperdocument models based upon the Rolodex card-index. Indeed, some popular authors have suggested that the essence of hypertext is the division of text into "chunks," a claim at odds with the ideals of Vannevar Bush, Douglas C. Engelbart, Nelson, Andries van Dam (see Carmody), and other hypertext pioneers. These researchers instead emphasized the support of full-sized documents, with all the complexities of authorial structure.

Even broad-based and influential essays on hypertext such as Jeff Conklin's "Hypertext" have often been read as supporting limited document models. An earlier version ("Survey") completely neglected intra- as opposed to inter-document links, which are crucial in any document of nontrivial size; it also ruled out links to locations within documents in general. Limitations that remained in the later article include requiring a one-to-one correspondence between windows and documents (which precludes some effective interface structures) and Conklin's devoting much discussion to methods for force-fitting documents into chunks while giving little motivation for such fitting except that many software systems impose it (35). Perhaps most unfortunate is that although Conklin states that the essence of hypertext is linking, not chunking, later references to his work have quoted primarily the parts that describe linking of directed graphs; at least when taken out of context these seem to imply that nodes in such graphs have no significant internal structure.

As with typewriter-model word processors, Rolodex-model hypertext systems have been warmly welcomed as a vast step forward, when in many ways they are more primitive than earlier systems. Indeed they provide many useful benefits and permit authors to create simple hypertexts easily and quickly; unfortunately, their too-concrete model leads to a fragmented computing environment of noncommunicating "hyper-islands" (Meyrowitz, "Cholesterol"). As the sophistication of document processing increases, so will perception of the limitations imposed by overly simplistic document models.

## Standards

### *Syntax: SGML and ODA*

Two major ISO standards exist for representing documents: SGML is in fact a metalanguage, in which a set of document-element classes, additional notations, and containment rules is declared, and a program can then parse and validate documents conforming to the declaration. SGML strongly encourages that the element classes constitute descriptive rather than procedural markup, though this is impossible to enforce fully. The Office Document Architecture (ODA) is more specific and provides a fixed set of descriptive-element classes and notations along with a representation of formatting. The standards' names reveal not only their designs but also their histories: SGML has been adopted for markup in a wide range of contexts because it easily describes a wide range of structures; ODA has been adopted for use in business contexts because its direct provision of formatting features is important there while the diversity and structural complexity of documents is lower.

The rapid spread of SGML exceeds anything we anticipated in 1987 and does not appear to be slowing. It is far from a perfect markup system, but it has been very effective in focusing attention on document structure rather than transient formatting and in facilitating document interchange.

### *Text Encoding: TEI*

Perhaps the most hopeful sign of all in the last five years has been that major text-encoding projects have committed themselves to a single, descriptive standard for marking up and interchanging the many documents they encode. As the Internet has made it convenient to pass around large documents, change them, enhance them, and send them back, it has also revealed the dangers of proprietary formats and permitted scholars to avoid them.

The Text Encoding Initiative (TEI) is a major international effort to provide guidelines for the encoding of machine-readable texts, particularly documents of interest to scholars. It is funded by the National Endowment for the Humanities, the European Economic Community, and the Mellon Foundation and has the support of many academic and professional societies. The efforts of many scholars during the first four years of the TEI's work have produced a volume of tagging guidelines that includes methods for marking up the detailed structure of literary prose and poetry, drama, multilingual texts, linguistic corpora, textual commentary and criticism of texts, links and temporal structures, and much more, including methods for creating extensions (see Sperberg-McQueen).

Several "affiliated projects" are preparing large bodies of text in accordance with the TEI guidelines. For example, the Women Writers' Project at Brown University is collecting, keying, and marking up all available works by female authors of English from 1350 to 1900. The encoding is particularly ambitious in that goals include producing a paper edition closely approximating the layout of the originals as well as a useful electronic edition. Other projects are encoding the works of philosophers such as Wittgenstein, Pierce, and Nietzsche.

The Perseus Project at Harvard University has encoded much of Classical Greek literature, plus a variety of secondary sources and thousands of linked images, again using SGML as the basis (see Mylonas). The database and the search and navigation tools for this database will be available before publication of this volume.

The computational linguistics community has found that electronic lexica continually diverge as various researchers restructure and annotate them, making it difficult to combine the enhancements made by different groups. This discovery led to the founding of the Dictionary Encoding Initiative, which is developing an SGML-based interchange format for natural language processing lexica and closely cooperates with TEI.

The entire Oxford English Dictionary has been encoded descriptively and is available on optical media with structure-aware search tools that support retrieval on the basis of etymologies, pronunciations, citations, dates, and many other features of the text (Tompa, **Clear**).

*Hypermedia Syntax: HyTime*

Although hypertext and hypermedia have not been so amenable to standardization as has text per se, common threads in the several standardization efforts have been

markup of structure as opposed to layout and frequently the use of SGML. Descriptive markup appears in a different guise for hypertext but, in short, involves marking up the fact and nature of the connection between linked objects, as opposed to the placement, form, or procedure for navigating among them.

A draft ISO standard called HyTime is now under consideration, which would provide methods for encoding links not only to text but to media, time-based phenomena, and anything else that can be modeled as a co-ordinate space (ISO). HyTime uses SGML as its syntactic basis, and many independent hypertext encoding efforts have chosen similar descriptive markup over previous proprietary forms.

### Barriers to the Future

It seems then that the trend is strongly toward the kinds of markup we advocated in 1987: descriptive markup of document structure, rather than procedural or presentational markup of ink structure. However, several issues remain, and these must be addressed by descriptive models as time goes on. A few of those issues are presented in this section.

#### *Tables, Equations, and Semi-textual Media*

For both tables and equations, some have rejected descriptive markup on the spurious ground that it imposes a silly interface, such as typing pointy-brackets directly:

```
<sum><from>i=0</from><to>n</to><of>i</of></sum>
```

In fact neither descriptive markup in general nor SGML in particular requires silly interfaces; at the same time, neither procedural nor presentational markup is immune to them. Precisely the same internal or interchange representation can be generated from any style of interface the user desires. The equation could be assembled graphically, via menus, or even via a natural language interface and displayed in whatever form is most convenient at any given moment:

```
The sum of i, taken from i=0 to n.
```

We defined descriptive markup in terms of what kinds of labels are associated with text portions: ones that describe the meaning or function of those portions, as opposed to ones that describe the appearance or particular processing. If we apply this same definition, descriptive markup for equations means marking them up for the meaning or function of each component as opposed to the layout. This is precisely the approach mathematicians seem to prefer, given the widespread use of AMS-TEX, which uses essentially this model (American Mathematical Society). Procedural, nondescriptive markup for the same sum shown above would be more as shown below, an interface that seems far less appealing than even the least attractive descriptive alternative:

```
stack(8pt times 'n'; 24pt greek 'S';
   8pt times 'i=0') i
```

Effective representations for tables and equations still need work. Although the hierarchical model used by SGML is superior to the flat "series of paragraphs" model that preceded it, tables and equations can only be treated as hierarchies at the cost of some naturalness. Nevertheless, descriptive markup itself, whatever its syntactic form, retains the advantages originally claimed.

*Name Spaces*

The capability to transfer and archive large collections of documents immediately raises the issue of how they can be named, so that the same document can be retrieved by the same name tomorrow or next year. This problem reaches beyond the question of how individual documents are encoded, into technological issues of file system and network topology and social and political issues of information access, publication, and copyright. It interacts closely with one issue raised first by Nelson in relation to hypertext: what happens when documents change?

*Version Management*

Historically, the act of print publication created a fixed form, which could not be changed although, of course, a new edition could be published in addition. The high cost of production, particularly when new printing plates were required, also made new editions major events. These realities have kept the number of editions of most works very small and therefore made the management of versions tractable.

Now that authors can change documents cheaply and easily and can redistribute the result, nothing guarantees fixed form. Electronic documents change without notice for reasons ranging from the author's correction of errors to a reader's desire to print with a particular device. The old standby of citing the date or number of an edition no longer suffices to distinguish the actual version to which reference is being made.

As documents change, possibly along divergent paths, unique and temporally persistent names for each version must be assigned and managed if references to them (hypertextual or not) are to remain valid. None of the current cataloging and identification schemes is prepared for this contingency. Library-catalog-numbering schemes do record version information but provide no means for verifying that a version is what it claims to be or has not changed subtly without the cataloguer's knowledge. Publisher's systems such as the ISBN, or international identifier registration mechanisms such as ISO Formal Public Identifiers, could be used to assign a unique name to each version, but they impose no principled way of determining that two objects with different names are versions of each other. Much work is needed in this area, developing reliable means for cataloging, naming, distinguishing, and validating the variant forms of electronic documents.

*Legal Vagueness*

The problem of versions is compounded by a lack of legal clarity over issues of copyright and intellectual property. At a time when the durability of document value is decreasing, new laws have extended the duration of copyright. Further, recent decisions appear to have extended the scope of copyright in ways the academic community (and much of the legal community) has found objectionable.

The vagueness in what constitutes copyrightable creative effort has led to great multiplication of text-encoding effort. Several firms sell expensive electronic copies of the King James Bible, dating from 1611, obviously a questionable practice. I count no fewer than five projects that have independently annotated the Greek New Testament with similar part-of-speech tags, plus one more that was narrowly averted by last-minute license concessions from another.

Many academic efforts have been compelled to encode older, less reliable texts of major authors because of inability to come to terms with those holding copyright over newer editions. Sometimes publishers put so high a price on electronic editions that scholars recreate their own from scratch, which decreases sales and sometimes

leads to a vicious cycle. In one recent incident scholars who were displeased with colleagues for failing to organize and release certain ancient texts quickly enough were moved to use computers to assemble the texts piece by piece from a concordance or inverted index and distribute that, to the dismay of some and delight of others (see Shanks).

The issues of copyright law are complex, bewilderingly so when their objects are in electronic form. However, it is of the utmost necessity to resolve them equitably, for the present waste of time, money, and effort is great indeed.

## Conclusions

A variety of factors—technological, economic, social, and legal—have affected the acceptance and actual use of advanced models of document structure in preference to relatively primitive ink-and-paper-oriented models. As scholars attempt more and more to pass their documents around, to co-author across networks, and to use their documents for multiple applications rather than only printing, they will demand increased portability, standardization, and flexibility of representation.

Just as in 1987, descriptive markup continues to offer a clear path to many of these goals. These few years have seen great advances in the availability, sophistication, and interface quality of systems. The problems that keep us from fully realizing these goals are increasingly economic and social rather than technological or theoretical in nature.

The seemingly inexorable move toward on-line literature raises profound questions about the nature of intellectual property, authorial control over not only the content but also novel applications of their documents, and publishers' and readers' roles. It is to be hoped that the next few years will bring serious consideration of such issues and perhaps some effective resolutions.

---

## Notes

1. The International Organisation for Standardisation (ISO) and a variety of industries are working together on a "universal character coding" standard, ISO 10646, which many hope will lead to a consistent solution to portability of data at this level.

2. See Meyrowitz, "Cholesterol," for a discussion of this proliferation.

**Bibliography**

American Mathematical Society. *AMS-TEX Version 2.1: User's Guide.* Providence: AMS, 1991.

Burkowski, Forbes J. "An Algebra for Hierarchically Organized Text-Dominated Databases." Manuscript. Department of Computer Science, U of Waterloo, 1991. (Portions previously appeared in "Intelligent Text and Image Handling," paper presented in Barcelona, Spain, April 1991 at RIAO '91.)

Bush, Vannevar. "As We May Think." Atlan*tic Monthly* 176 (July 1945): 101-08.

Carmody, Steven, Walter Gross, Theodor H. Nelson, David Rice, and Andries van Dam. "A Hypertext Editing System for the /360." *Pertinent Concepts in Computer Graphics.* Ed. M. Faiman and J. Nievergelt. Urbana: U of Illinois P, 1969. 291-330.

Catano, James. "Poetry and Computers: Experimenting with the Communal Text." *Computing and the Humanities* 13 (1979): 269-75.

Conklin, Jeff. "A Survey of Hypertext." MCC Technical Report Number STP-356-86, Rev. 1. Austin: Microelectronics and Computer Technology Corporation, 1986.

_____. "Hypertext: An Introduction and Survey." *IEEE Computer* 20 (1987): 17-41.

DeRose, Steven J., David G. Durand, Elli Mylonas, and Allen H. Renear. "What Is Text, Really?" *Journal of Computing in Higher Education* 1(1990): 3-26.

Engelbart, Douglas C., and William K. English. "A Research Center for Augmenting Human Intellect." 1968. *Computer-Supported Cooperative Work: A Book of Readings.* Ed. Irene Greif. San Mateo: Morgan Kaufmann, 1988. 81-105.

Feiner, Steven, S. Nagy, and Andries van Dam. "An Integrated System for Creating and Presenting Complex Computer-Based Documents." *Computer Graphics* 15 (August 1981): 181-89.

International Organisation for Standardisation. ISO 8879. *Information Processing—Text and Office Systems—Standard Generalized Markup Language (SGML).* 1st ed.—1986-10-15. Ref. No. ISO 8879-1986.

_____. "ISO/IEC DIS 10744: Hypermedia/Time-based Structuring Language: HyTime." Draft. 1991.

Killough, Ronnie Lynn. "Hypertext Interchange with the Dexter Model: Intermedia to KMS." Master's Project. Department of Computer Science, Texas A and M U, August 1990.

Meyrowitz, Norman. Closing address at Hypertext '87. Chapel Hill, 13-15 November 1987.

_____. "Hypertext—Does It Reduce Cholesterol, Too?" Keynote address at Hypertext '89. Pittsburgh, 6 November 1989. (Reprinted in *From Memex to Hypertext: Vannevar Bush and the Mind's Machine.* Ed. James M. Nyce and Paul Kahn. Boston: Academic P, 1992. 287-318.)

Mylonas, Elli, Gregory Crane, Kenneth Morrell, and D. Neel Smith. "The Perseus Project: Data in the Electronic Age." *Computing and the Classics.* Ed. J. Solomon and T. Worthen. Tempe: U of Arizona P, forthcoming.

Nelson, Theodor H. *Computer Lib/Dream Machines.* 1974. Seattle: Microsoft P, 1987.

Prusky, John. *FRESS Resource Manual.* Providence: Brown U Department of Computer Science, 1978.

Reid, Brian K. *Scribe: A Document Specification Language and Its Compiler.* Dissertation. Pittsburgh: Carnegie-Mellon U, 1980. (Available as Technical Report CMU-CS-81-100.)

Shanks, Hershel. "Why Transcripts of the Dead Sea Scrolls Had to Be Released." *Chronicle of Higher Education* 38 (2 October 1991): B1-B2.

Sperberg-McQueen, C. Michael, and Lou Burnard. *Guidelines for the Encoding of Machine-Readable Texts.* Chicago: Text Encoding Initiative, 1989.

Tompa, Frank W. "What Is (Tagged) Text?" *Dictionaries in the Electronic Age: Proceedings of the Fifth Annual Conference of the UW Centre for the New Oxford English Dictionary.* 1989. 81-93.